ELEMENTS OF COMPUTING

PROJECT REPORT-COMPILER-I

Eric Oommen Mathew

AMENU4AIE21027

Classes and functions used: -

1. **Compiler**

    1.1. 1It analyses grammar

    1.2. Takes input from Tokeniser

    1.3. Outputs to the output file

    1.4. Routines

        1.4.1. Compile statements

        1.4.2. Compile ‘if’ statements

        1.4.3. Compile ‘while’ statements

    1.5. API

        1.5.1. constructor: - To create a new compilation

        1.5.2. compile class: - Compiles a complete class

        1.5.3. compileClassVarDec: - Declares static variables

        1.5.4. compileSubroutinDec: - Methods function

        1.5.5. compileParameterList : - Empty parameter list

        1.5.6. compileSubroutinBody: - subroutine’s body

1.5.7. compileVarDec: - Variable declaration

1.5.8. compileStatements: - Sequence of statements

2. Jack analyser

2.1. Main module

2.2. Takes in jack files

2.3. Outputs xml file

3. Jack tokenizer

3.1. String constants are outputted without double quotes

3.2. <,>,",and & are outputted as $lt; > " and &lt

3.3. Listing out the tokens

3.4. Functions

3.4.1. Symbol constructor

3.4.2. "ContainSymbol" constructor

3.4.3. advance()

3.4.4. tokenType()

3.4.5. keyword()

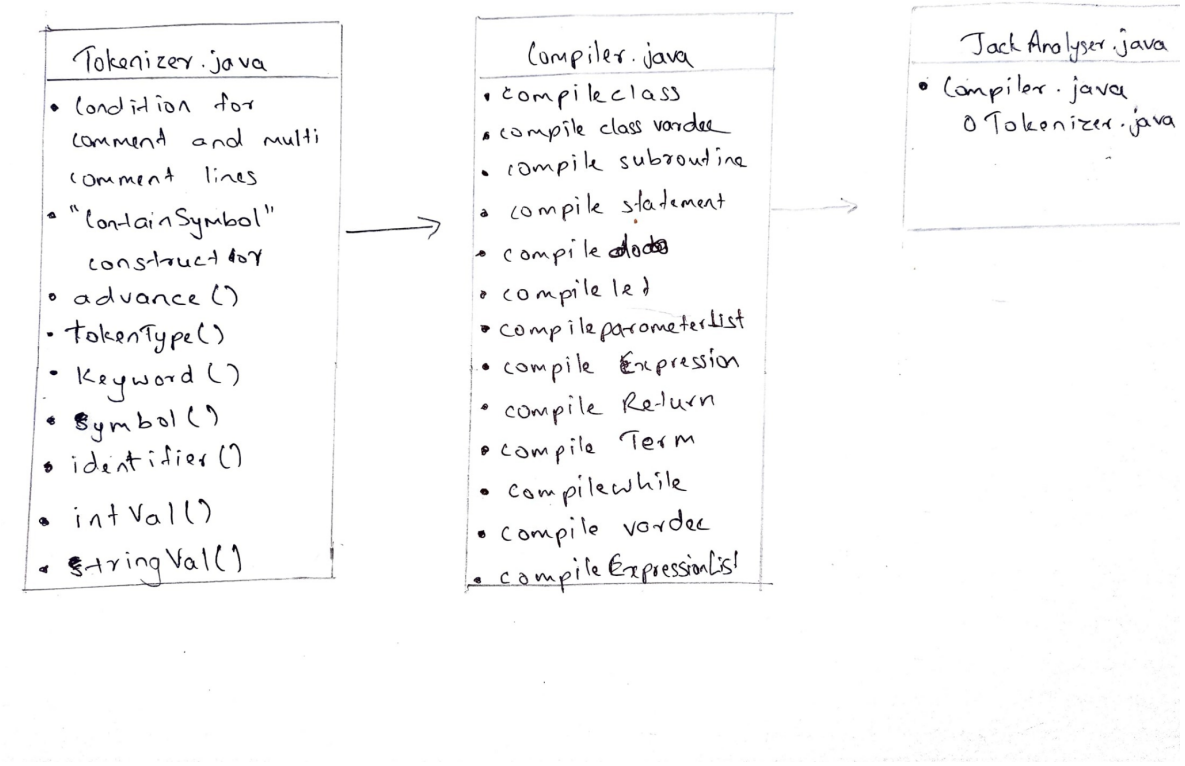
3.4.6. symbol()

3.4.7. identifier()

3.4.8. intVal()

3.4.9. stringVal()

Class diagram: -



My contribution to the project: -

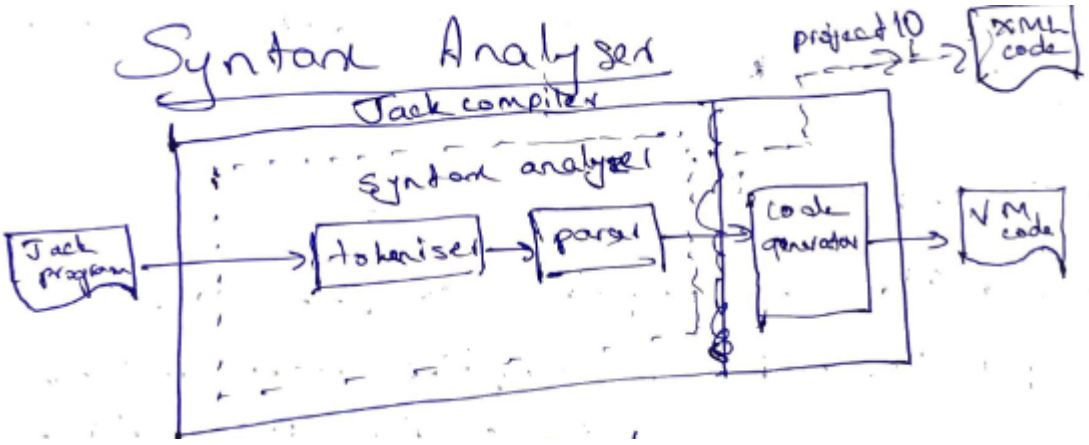
I and Manav did Jack Tokeniser

Jack Tokenizer: -

1. String constants are outputted without double quotes
2. <,>,",and & are outputted as $lt; > " and &lt
3. Listing out the tokens
4. **Handling expressions**
    a. When the current token is a varName(some identifiers),it can be either a variable,name,an array entry,or a subroutine call

b. To resolve which possibility we are in,The parser should”look ahead”,save the current token,and advance to next one

c. There is no compileSubroutineCall method, rather the subroutine call logic is handled in compile Term



Insights from the course: -

The course helped us to build an idea about the modern software hierarchy

The course taught us about the basics of Boolean algebra and further help us to understand the working of the Modern Computer .

We learned about all the logic gates and about the software nand2tetris;We learned about different types of memory

We got introduced to A and C instruction.

We learnt about assembler and how to translate high level code to assembly language by handling instructions, variables and symbols with the help of the construction of the Symbol table .

Introduction to Virtual Machine, got introduced to stack data structure and learned about two functions push and pop. We learnt stack machines can be manipulated by arithmetic/logical commands, memory segments command branching commands and function commands We learnt about function call and return.

Then we learned about a new high level language Jack got familiarised with the syntax

Then we learned about compiler and learnt about tokenizer and its functions got introduced to jack Grammar and build a jack tokenizer

Command to invoke syntax analyser: -

```
public class Jack_Analyzer {
    public static void main(String[] args) throws FileNotFoundException, IOException {
        Compiler compile = new Compiler("D:\\programs\\Eric Oommen\\AM.EN.U4AIE21071\\nand2tetris\\projects\\10\\Square\\Main.jack");
    }
```